ChungPy Documentation

Release 1.0.1

Corinna Ernst

January 19, 2015

Contents

An implementation of Chung's linear time algorithm for solution of the Maximum Density Segment Problem with additional features, namely simultaneous identification of Minimum Density Segments and consideration of content constraints

Given a sequence S of n numerical pairs (a_i, w_i) with values a_i and widths $w_i > 0$ for $i \in 1, ..., n$ the width of a consecutive subsequence $S(i, j)$ of S with $1 \leq i \leq j \leq n$ is given by

$$w(i,j) = \sum_{k=i}^{j} w_k$$

and its density is given by

$$d(i,j) = \frac{\sum_{k=i}^{j} a_k}{w(i,j)}.$$

The Maximum Density Segment Problem (MDSP) comprises the identification of a consecutive subsequence $S(i^*, j^*)$ with largest possible density, i.e. the Maximum Density Segment.

Chung & Lu [1] presented a linear time algorithm for solution of the MDSP with arbitrary width contraints w_{min} and w_{max}, s.t. $w_{min} \leq w(i^*, j^*) \leq w_{max}$. Their algorithm returns for each possible stop index j' of possible subsequences $S(i', j')$ a corresponding start index $i_{j'}$, s.t. $i_{j^*} = i^*$.

ChungPy is an implementation of Chung & Lu's algorithm including the ability of additional consideration of content constraints c_{min} and c_{max}, s.t. $c_{min} \leq j^* - i^* + 1 \leq c_{max}$. Furthermore, ChungPy allows for computation of Minimum Density Segments, i.e. the identification of a consecutive subsequence $S(i^{**}, j^{**})$ with smallest possible density.

ChungPy is hosted at https://bitbucket.org/corinnaernst/chungpy.

Contents:

[1] K.-M. Chung and H.-I. Lu. An optimal algorithm for the maximum-density segment problem. SIAM Journal on Computing, 34(2), 2005.

CHAPTER 1

Chunpy: How-to

Use module `chungpy.mdsp` for computation of Maximum Density Segments exclusively.

Input persists of at least value vector *a* and width vector *w* which can be of type *list* or numpy *array*.

Note, that *a* and *w* have to be of identical length and all elements of *w* have to be greater than 0.

The corresponding MDSP is solved in concurrence with the instantiation of the problem instance.

```
import chungpy

m=chungpy.mdsp([1,-5,12,7,-12], [2,14,3,7,6])
print(m)
```

yields the following output

```
# Overview
a = [1.0, -5.0, 12.0, 7.0, -12.0]
w = [2.0, 14.0, 3.0, 7.0, 6.0]
length = 5
min/max content = 1,5
min/max width = 1,inf

# Maximum Density Segment per Position
1 -> 1, c=1, w=2.0, d=0.5
1 -> 2, c=2, w=16.0, d=-0.25
3 -> 3, c=1, w=3.0, d=4.0
3 -> 4, c=2, w=10.0, d=1.9
3 -> 5, c=3, w=16.0, d=0.4375

# Maximum Density Segment(s)
3 -> 3, c=1, w=3.0, d=4.0
```

Define width constraints w_{min} and w_{max} via attributes `min_width` and `max_width` and content constraints c_{min} and c_{max} via attributes `min_cont` and `max_cont`.

```
m=chungpy.mdsp([1,-5,12,7,-12], [2,14,3,7,6], max_width=20, min_cont=2)
print(m)
```

yields the following output

```
# Overview
a = [1.0, -5.0, 12.0, 7.0, -12.0]
w = [2.0, 14.0, 3.0, 7.0, 6.0]
length = 5
min/max content = 2,5
```

```
min/max width = 1,20

# Maximum Density Segment per Position
1 -> 2, c=2, w=16.0, d=-0.25
1 -> 3, c=3, w=19.0, d=0.42105263157894735
3 -> 4, c=2, w=10.0, d=1.9
3 -> 5, c=3, w=16.0, d=0.4375

# Maximum Density Segment(s)
3 -> 4, c=2, w=10.0, d=1.9
```

The set of stop indices of maximum density segments can be accessed via attribute `result_inds` and the corresponding maximum density can be accessed via `max_dens`.

Local maximum density segments, i.e. start indices of the shortest segments with highest possible density stopping at each posssible stop index j' can be accessed via attribute `result_inds` of type `dict`. Of course, the keys in `result_inds` does only contain stop indices of segments fullfilling the given width and content constraints.

Width and density of arbitrary segments of the given problem instance can be computed via methods `width` and `density`. Note, that start and stop indices are 1-based and stop index j is inclusively.

The following code snippet returns the densities of all local maximum density segments of MDSP instance `m`:

```
[m.dens(m.result_inds[j],j) for j in m.result_inds]
```

Use module `chungpy.mmdsp` for simultaneous computation of Maximum and Minimum Density Segments.

```
import chungpy

m=chungpy.mmdsp([1,-5,12,7,-12], [2,14,3,7,6])
print(m)
```

yields the following output

```
# Overview
a = [1.0, -5.0, 12.0, 7.0, -12.0]
w = [2.0, 14.0, 3.0, 7.0, 6.0]
length = 5
min/max content = 2,5
min/max width = 1,20

# Maximum Density Segment per Position
1 -> 2, c=2, w=16.0, d=-0.25
1 -> 3, c=3, w=19.0, d=0.42105263157894735
3 -> 4, c=2, w=10.0, d=1.9
3 -> 5, c=3, w=16.0, d=0.4375

# Maximum Density Segment(s)
3 -> 4, c=2, w=10.0, d=1.9

# Minimum Density Segment per Position
1 -> 2, c=2, w=16.0, d=-0.25
2 -> 3, c=2, w=17.0, d=0.4117647058823529
3 -> 4, c=2, w=10.0, d=1.9
4 -> 5, c=2, w=13.0, d=-0.38461538461538464

# Minimum Density Segment(s)
4 -> 5, c=2, w=13.0, d=-0.38461538461538464
```

The set of stop indices of minimum density segments can be accessed via attribute `min_result_inds` and the

corresponding minimum density can be accessed via `min_dens`.

Local minimum density segments, i.e. start indices of the shortest segments with lowest possible density stopping at each posssible stop index j' can be accessed via attribute `min_result_inds` of type `dict`.

CHAPTER 2

Chunpy API Documentation

class chungpy.mdsp.**mdsp**(*vals*, *widths*, *min_width=None*, *max_width=None*, *min_cont=None*, *max_cont=None*)

Maximum Density Segment Problem Instance

Initiation of problem instance with concurrent solution of the corresponding Maximum Density Segment Problem

Parameters

- **vals** (*list of numerical values*) – list of values
- **widths** (*list of numerical values*) – list of width values
- **min_width** (*numerical*) – minimum width of maximum density segments (default 1)
- **max_width** (*numerical*) – maximum width of maximum density segments (default sum(widths))
- **min_cont** (*int*) – minimum content, i.e. length, of maximum density segments (default 1)
- **max_cont** (*int*) – maximum content, i.e. length, of maximum density segments (default len(vals))

vals
list of numerical values

list of values

widths
list of numerical values

list of width values

length
int

length of vals, i.e. length of widths

min_width
numerical

minimum width of maximum density segments (default 1)

max_width
numerical

maximum width of maximum density segments (default sum(widths))

min_cont
int

minimum content, i.e. length, of maximum density segments (default 1)

max_cont
int

maximum content, i.e. length, of maximum density segments (default len(vals))

result_inds
dict of ints

mapping of stop indices to the corresponding start index of the corresponding maximum density segment

max_dens
float

density of the overall maximum density segment

max_inds
set

stop positions of all overall maximum density segments

dens(*start*, *stop*)
Return the density of segment with given constraints.

Parameters

- **start** (*int*) – index of segment's start position (1-based)
- **stop** (*int*) – index of segments's stop position (1-based)

Returns **density** – density of segment from start to (inclusively) stop

Return type float

width(*start*, *stop*)
Return the width of segment with given constraints.

Parameters

- **start** (*int*) – index of segment's start position (1-based)
- **stop** (*int*) – index of segments's stop position (1-based)

Returns **width** – width of segment from start to (inclusively) stop

Return type int or float

class chungpy.mmdsp.**mmdsp**(*vals*, *widths*, *min_width=None*, *max_width=None*, *min_cont=None*, *max_cont=None*)
Maximum and Minimum Density Segment Problem Instance

Initiation of problem instance with concurrent solution of the corresponding Maximum AND Minimum Density Segment Problem

Parameters

- **vals** (*list of numerical values*) – list of values
- **widths** (*list of numerical values*) – list of width values
- **min_width** (*numerical*) – minimum width of maximum density segments (default 1)
- **max_width** (*numerical*) – maximum width of maximum density segments (default sum(widths))

- **min_cont** (*int*) – minimum content, i.e. length, of maximum density segments (default 1)
- **max_cont** (*int*) – maximum content, i.e. length, of maximum density segments (default len(vals))

vals
list of numerical values

list of values

widths
list of numerical values

list of width values

length
int

length of vals, i.e. length of widths

min_width
numerical

minimum width of maximum density segments (default 1)

max_width
numerical

maximum width of maximum density segments (default sum(widths))

min_cont
int

minimum content, i.e. length, of maximum density segments (default 1)

max_cont
int

maximum content, i.e. length, of maximum density segments (default len(vals))

result_inds
dict of ints

mapping of stop indices to the corresponding start index of the corresponding maximum density segment

max_dens
float

density of the overall maximum density segment

max_inds
set

stop positions of all overall maximum density segments

min_result_inds
dict of ints

mapping of stop indices to the corresponding start index of the corresponding minimum density segment

min_dens
float

density of the overall minimum density segment

min_inds
set

stop positions of all overall minimum density segments

CHAPTER 3

Indices and tables

- *genindex*
- *modindex*
- *search*

C